Collaborative Filtering

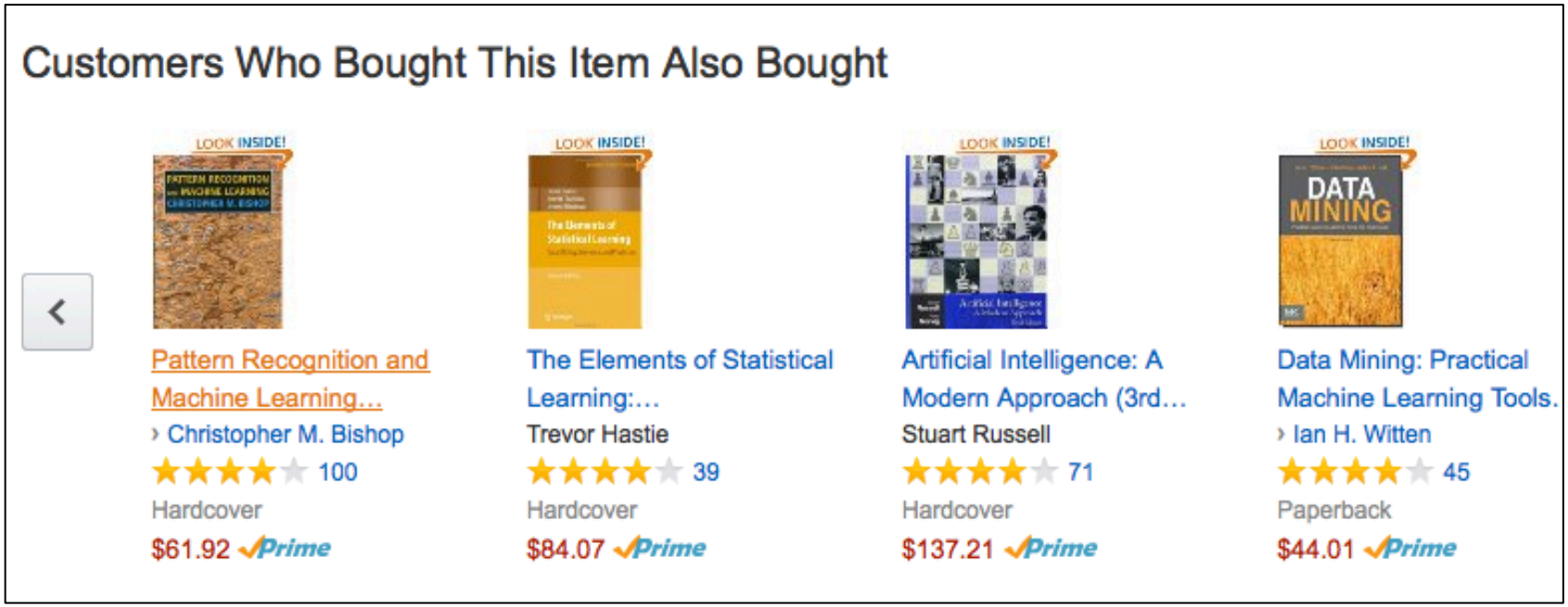
EECS 349 Machine Learning Recitation

Bongjun Kim

Oct. 10, 2014

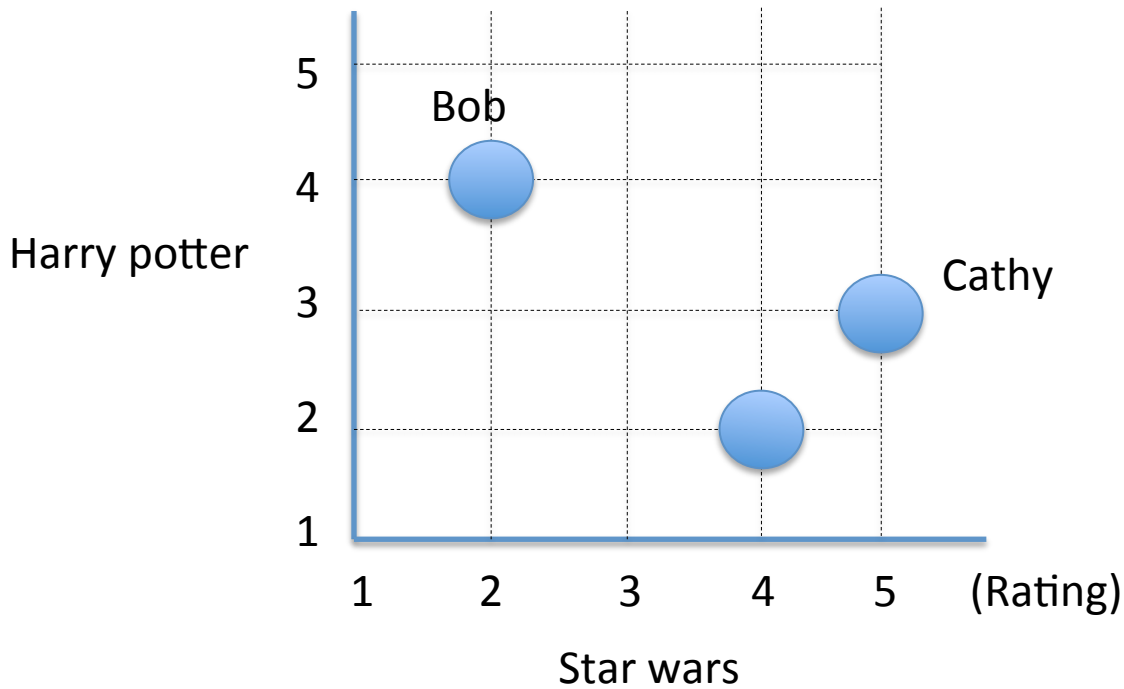
Collaborative Filtering

- Recommendation system
  - Search for a user that is similar to you
  - You might like the item the user likes.



How do we find a user who is similar?

- Distance (or similarity) measure
  - N-dimensional space



Which similarity measure to use?

- p-norm
  - Manhattan
  - Euclidian
- Pearson Correlation
- Cosine Similarity

Who is the most similar to John?

Example #1

	Inception	Begin again	Once
Brian	5	2	2
Bob	1	4	4
Cathy	2	3	3
John	5	1	2

- Manhattan Distance:

(John, Brian) = 0 + 1 + 0 =1
(John, Bob) = 4 + 3+ 2 =9
(John, Cathy) = 3 + 2 + 1 = 6

Q: Does Manhattan Distance measure similarities properly in this data set?

Who is the most similar to Adam?

Example #2

	Inception	Begin again	Once	Star wars
Bill	2	3	3	2
Brian	5	1	1	5
Jane	5	1	2	4
Adam	3	2	2	3

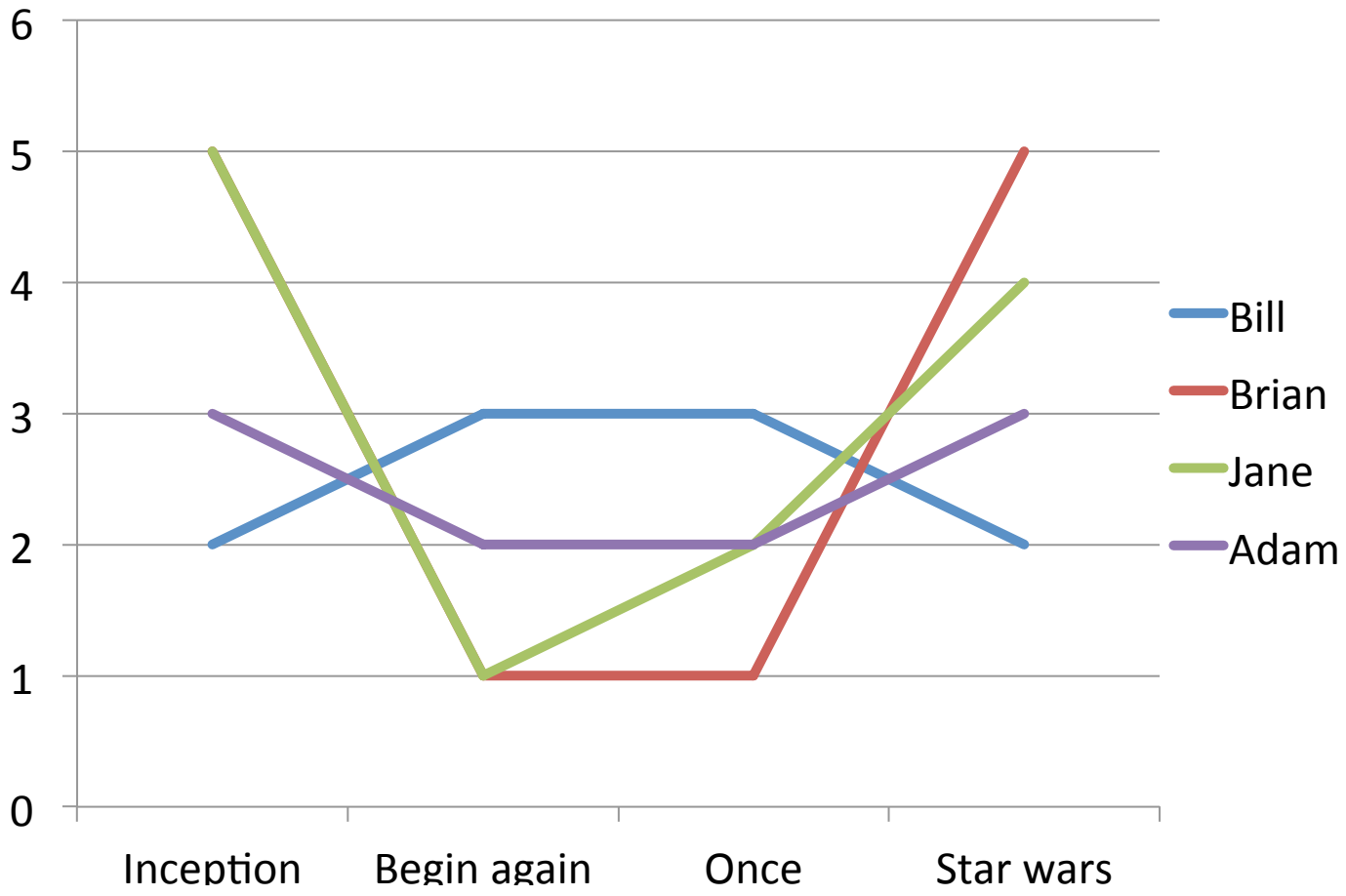
- Manhattan Distance:

(Adam, Bill) = 1 + 1 +1 + 1 =4
(Adam, Brian) = 2 + 1+ 1 + 2 = 6
(Adam, Jane) = 1 + 1 + 1 + 1= 4

Q: Does Manhattan Distance measure similarities properly in this data set?

Different users may use different rating scales

Let’s see correlations between users



Pearson Correlation

- Measure of correlation between two variables
- Pearson correlation coefficient
  - Range (-1, 1)
  - 1 indicates perfect correlation.

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i \in C}(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

```
>> import scipy.stats
>> scipy.stats.pearsonr(array1, array2)
```

Who is the most similar to Adam?

Example #2

	Inception	Begin again	Once	Star wars
Bill	2	3	3	2
Brian	5	1	1	5
Jane	5	1	2	4
Adam	3	2	2	3

- Pearson Correlation:

(Adam, Bill) = -1
(Adam, Brian) = 1
(Adam, Jane) = 0.94

Q: Does Pearson Correlation measure similarities properly in this data set?

How to predict ratings to unrated items

- K- Nearest Neighbor Collaborative Filtering
  - Pick k users that had similar preferences to those of current user
  - Factors the relative proximity of k nearest neighbors
  - You need to do experiments to find optimal k value.

Let's practice k-NN CF (k=1)

Example #1

	Inception	Begin again	Once	Star wars
Brian	5	2	2	4
Bob	1	4	4	2
Cathy	2	3	3	1
John	5	1	2	?

Manhattan Distance:
(John, Brian) = 0 + 1 + 0 =1
(John, Bob) = 4 + 3+ 2 =9
(John, Cathy) = 3 + 2 + 1 = 6

Let's practice k-NN CF (k=1)

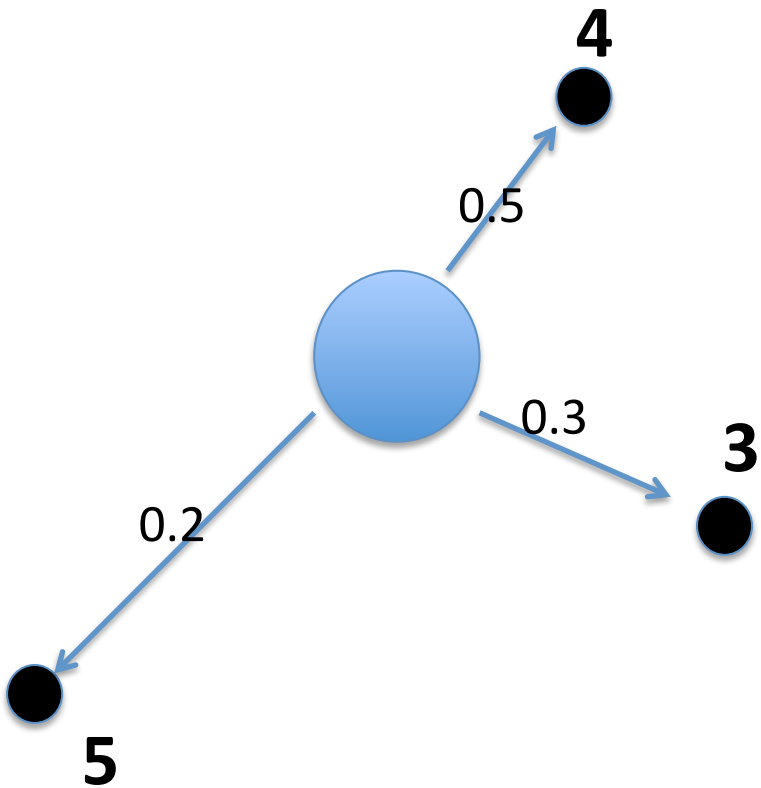
Example #2

	Inception	Begin again	Once	Star wars	Avatar
Brian	2	3	3	2	4
Bob	5	1	1	5	2
Cathy	5	1	2	4	1
John	3	2	2	3	?

Manhattan Distance:
(John, Brian) = 1 + 1 +1 + 1 =4
(John, Bob) = 2 + 1+ 1 + 2 = 6
(John, Cathy) = 1 + 1 + 1 + 1= 4

Pearson Correlation Coefficient
(John, Brian) = -1
(John, Bob) = 1
(John, Cathy) = 0.94

When K >= 2



Item-based CF

Example #1

	Inception	Begin again	Once
Brian	5	2	2
Bob	1	4	4
Cathy	2	3	3
John	5	1	2

- Manhattan Distance:

(Once, Inception) = 3 + 3 +1+3 =10

(Once, Begin again) = 0+0+0+1 =1

Let's practice k-NN CF (k=1)

Example #1

	Inception	Begin again	Once	Star wars
Brian	5	2	2	4
Bob	1	4	4	2
Cathy	2	3	3	1
John	5	1	2	?

Manhattan Distance:

(Star wars, Inception) = 1 + 1 + 1 =3

(Star wars, Begin again) = 1 + 2+ 2 =5

(Star wars, Once) = 2+2+2 = 6

Dealing with missing values

Example #2

	Inception	Begin again	Once	Star wars	Avatar
Brian	2	?	3	?	4
Bob	5	1	1	5	2
Cathy	5	?	2	2	1
John	5	?	2	3	?

Dealing with missing values

Example #2

	Inception	Begin again	Once	Star wars	Avatar
Brian	2	0	3	0	4
Bob	5	1	1	5	2
Cathy	5	0	2	2	1
John	5	0	2	3	?

Make a decision

- Which distance measure to use?
- How many neighbors to pick?
- How to weight neighbors chosen?
- User-based or item-based?
- How to deal with missing values?
  - Discard data related to unrated items?
  - Fill Zero or average value?
  - Other advanced imputation technique?